The Ariadne String against COVID -19 pandemic propagation: outdoor path selection with limited virus exposure

Philippe Jacquet, Liubov Tupikina

The Ariadne String against COVID -19 pandemic propagation: outdoor path selection with limited virus exposure

Philippe Jacquet
Inria, Saclay Ile de France
France
philippe.jacquet@inria.fr
Liubov Tupikina
Nokia Bell labs, France, Center for Research and Interdisciplinarity (CRI),
Université de Paris – INSERM (U1284), 75004 Paris, France
liubov.tupikina@nokia-bell-labs.com

***Abstract*—The present work is an extension of [1] where local outdoor excursion was analysed and optimized in order to limit the exposure to virus during city lock-downs. The present work is applicable for analysis of situations after lock-down where outdoor excursions have a aim (commuting to work or shopping) and may not be strictly local. In particular we investigate the biking path optimization. We use drifted random walks for this optimization which under the condition of uniform distribution of initial starting points and destination locations gives a perfect or quasi perfect load balancing of streets. We compare this with the case when biking lanes are just following the main commuting lanes by bus or subway, thus minimizing the exposure to virus by a factor between 3 and 9 on various models. Interestingly the path selection is immediately beneficial to the first user even if the other users stay on preferential paths.**

I. Introduction

The COVID-19 pandemic has forced more than half of mankind into lock-down situation [5], [9], [8]. During lock-down in some countries outdoor excursion were authorized under strict restrictions; in France, no more than 1 hour less than 1km away from home. These restrictions had an aim to reduce the exposure rate to virus through outdoor contact with other persons. This problem is crucial in urban places, since in country places the distinction are easier to achieve. In [1] we have described an application, called Ariadne String against COVID, which reduces the contact rate thanks to a better load balancing streets while avoiding crowded areas. The fundamental tool in Ariadne COVID is the use of random walk which naturally loads a balance edges in a graph. The application is shown to give a benefit to the first user, although to have an impact on the pandemic, it must be used by a majority of users. The average outdoor exposure rate reduction is around 3, but it is difficult to verify this claim since it is based on pre-lock-down partial data on some cities.

In the present work we present Ariadne-2 algorithm as a follow-up application whose aim is to limit outdoor exposure to virus *after* lock-downs. The lock-down's aim is to "flatten the curve" in order to limit the congestion in hospitals. It is equivalent to the graphite bars used to slow down and reverse the reactions in the kernel of a nuclear plant. If you remove the graphite bars, the reaction will restart. Therefore the social distancing and other ways to limit exposure will need to be continued after lock-down before either a vaccine or a herd-immunity is developed.

But contrary to lock-down outdoor excursions which were set-up for healthy consideration and therefore were aimless (walking, jogging), the outdoor excursions after lock-down will have an aim, most likely in order to commute between home and work place. The public transportation is considered to be unsafe since people will be packed in small areas, and it should be restricted in order to achieve a sufficient social distances. The city of Paris envisions to help people to rely on biking and other surface transportation by increasing the number of streets with biking lanes. However the biking lanes are narrow and sometimes very busy because people commute around the same hours. Our work is to introduce a policy which helps the user to find a biking route from home to work which limit the exposure rate by achieving a load balancing between lanes. The exposure rate can be reduced by a factor ranging from 3 to 9 depending on the model.

Technically a simple random walk model does not work as well, since it will take an eternity for a homogeneous random walk to connect the home address with the work place, if the latter location is a random but fixed location in the city. The shortest path will not be good, since it is likely to take the main streets and create a gathering on those streets. Our approach is based on *drifted* random walks so that the penalty with the shortest path will be limited, while the aggregation of path from all users will give an optimal load balancing over the streets and therefore minimize the exposure rate.

Here we suggest the improved algorithms of drifted random walks and validate them on two models of a city map: the grid models and the Voronoi tessellation model. The parameters of these models are inspired of the parameter of the city of Paris. We have also applied the algorithm to the real streets data

using openstreetmap API [13]. Interestingly we have worked on cities where the map contains many obstacles such are rivers and swamp areas which were not in our previous models. We have chosen the Kaliningrad-Königsberg area famous for its complex network of bridges since Euler.

Interestingly the algorithms are less CPU consuming than classic shortest path algorithm and therefore can be implemented over light servers

II. Description of the Ariadne-2 algorithm

The main algorithm is inspired from the opportunistic routing [2] and the geo-routing protocol [3], which uses the information about coordinates of nodes destinations. It works as follows. The user sends its GPS coordinate $z_0 = (x_0, y_0)$ of its home address (her/his initial position). She/he also sends the coordinate of her/his final destination $z_f = (x_f, y_f)$. The algorithm operates on an abstract graph, which may represent a dataset consisting of street intersection (node) linked by a segment of a street (edge). The intersection I is given by its GPS coordinates $(x(I), y(I))$. Two intersections I and J are connected if there exist a segment of street which connects two intersections. Let V the set of intersections and E the set of street segments, The pair (V, E) forms a graph. We also have the set S of streets, A street is a set of contiguous street segments. All neighbors of an intersection I are the list of intersections to which I is connected. We use a number $\epsilon > 0$ fixed for the protocol.

An initial position of a user is on a segment $(I_U, I_D) \in E$. The path will take the intersection among I_U and I_D which lies ahead of a destination, *i.e* if $\langle z_f - z_0 | I_U - I_D \rangle \geq 0$ then I_U is selected: $z_1 = I_U$, otherwise $z_1 = I_D$.

At step number k, let z_k be the intersection, where the path is currently ending. Let d be the degree of the intersection z_k. If $d = 1$, then the path backtracks. If $d > 1$ then with probability ϵ the path proceeds as a non-backtracking random walk, *i.e.* the path takes any other neighbor edge distinct of (z_{k-1}, z_k) with equal probability. Otherwise, with probability $1-\epsilon$, the path takes one of the two best sectored edges, defined as follows. Assuming the angle of the edge (z_k, z_{k-1}) is θ_0, and we enumerate $\theta_1, \theta_2, \dots \theta_{d-1}$ the angle in increasing order of the other edges of intersection z_k. Let θ be an angle of the vector (z_k, z_f) toward the destination.

Let j be such that $j\frac{2\pi}{d} < \theta \leq (j+1)\frac{2\pi}{d}$. Note here that $j+1$ must be considered by *modulo* of d. Let variables $\alpha_d(\theta)$ and $\beta_d(\theta)$ such that $\alpha_d(\theta) + \beta_d(\theta) = 1$ and $\alpha_d(\theta)e^{2ij\pi/d} + \beta_d(\theta)e^{2i(j+1)\pi/d}$ be proportional to $e^{i\theta}$. Numerically we have

$$\begin{cases} \alpha_d(\theta) &= \frac{\sin(2(j+1)\pi/d-\theta)}{\sin(\theta-2j\pi/d)+\sin(2(j+1)\pi/d-\theta)} \\ \beta_d(\theta) &= \frac{\sin(\theta-2j\pi/d)}{\sin(\theta-2j\pi/d)+\sin(2(j+1)\pi/d-\theta)}. \end{cases} \quad (1)$$

The selection of the next step z_{k+1} is done as follows: with probability $\alpha_d(\theta)$ it selects the edge corresponding to angle θ_j, and with probability $\beta_d(\theta)$ the edge corresponding to the angle θ_{j+1}.

We notice that there is no reason that the selected edge actually heads toward the destination (in fact it will never do), but on average it will do, under mild conditions on the actual street angles distribution in the city.

Notice that when θ_0 is the actual angle selected by the path algorithm selection, then the path backtracks. In this case the loop removal should be applied to the actual path, this will even more reduce the actual weight of the path.

The algorithm is much simpler than the actual Dijkstra shortest path algorithm. Indeed the shortest path algorithm leads to at least an average quadratic complexity in terms of the number of vertices, while the randomized algorithm is at most linear, in fact proportional to the diameter of the graph. This will greatly help for the computational power required to have a server capable of serving several millions of requests. However we shall not expect the algorithm to provide the shortest path, however we expect that the randomized algorithm will provide a reasonable penalty for having no optimal path.

A. Types of algorithms

All in all, here we work with four different types of algorithms:

1) the shortest path algorithm;
2) the preferential path algorithm;
3) the isotropic walk algorithm;
4) the geo-routing algorithm.

The three last algorithms perform the same on a grid network, this is why we did not distinguish them in the previous section. The isotropic walk algorithm is the main algorithm we focus in this paper. Note that the shortest path algorithm is the most expensive of all three since its quadratic in the number of nodes and therefore maybe expensive in term of server complexity.

B. Isotropic walk algorithm

Definition 1 (Isotropic walk condition)**.** *A walk is isotropic at a given intersection I, if when reaching an intersection I, the difference of an angle θ toward a destination with an angle θ_0 to an arriving edge, satisfies a fixed distribution $P_I(\theta - \theta_0)$.*

We can show that under the *isotropic walk* condition, defined below, the algorithm leads to an uniform distribution.

Notice that the walk arriving at a destination has stochastic component assuming the randomness of the initial and destination coordinates. Notice that the isotropic condition does not imply that $\theta - \theta_0$ must always be close to π (most direct path) and when $\theta - \theta_0$ is small the path will most likely backtrack.

Theorem II.1. *Under the isotropic walk condition, and assuming the same constant speed v of all travellers, the aggregation of paths leads to uniform densities of travellers on streets.*

Proof. We use the classic proof of stationary random walks in undirected graphs. Let us assume that at time t every street in a city has the same exit rate $\rho(t)$ on each of its end points. We will prove that all streets have the same entrance rate. And consequently the same exit rate on the other end. Thus the uniform distribution is the stationary distribution of the entrance rates.

In the following we assume that there is no loss b entrance and exit rates, assuming for example that initial and destination points of a path balance in all stre equivalently that an arrival point coincides with a de point like in a random way point mobility model.

Let consider a traveller arriving at an intersection I, has d arriving streets. According to the uniformization thesis, the traveller has an equal probability to arrive on the d intersecting edges. With probability ϵ it selects any $d-1$ other streets. With probability $1-\epsilon$ it select one of t sectored edge. Enumerating the edges in counter clo way, initialising with the entrance edge, the isotropi conditions leads to the following expression of the probability $p_d(j)$ that the edge j is selected is

$$p_d(j) = \int_{2j\pi/d}^{2(j+1)\pi/d} P(\theta)\alpha_d(\theta) + \int_{2(j-1)\pi/d}^{2j\pi/d} P(\theta)\beta_d(\theta). \quad (2)$$

Therefore the probability that an edge is selected, independently of the entrance edge is

$$\frac{d-1}{d}\epsilon + (1-\epsilon)\sum_{j=0}^{j=d} p_d(j). \quad (3)$$

Thus the exit rates are uniform, and have value $\rho(t)$ and consequently the stationary rate distribution is uniform with some value ρ. Since the speed is considered to be the same on each street, the density of travellers on each street is ρ/v per unit length. □

As an intermediate step, there is the randomized geo-routing algorithm, which is very close to the isotropic walk with the difference that instead of assuming equal angular sectors $j\frac{2\pi}{d}$ for the exit streets at each intersection, we take the actual angle of streets. The consequence of that is twofold: (i) the average exit path is now well aligned with the angle toward the destination, (ii) we lose the isotropic property, since the densities on the exit street will now vary with the variation of angles between the streets.

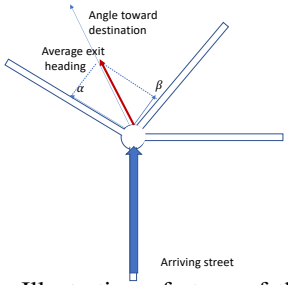

Figure 1: Illustration of steps of the randomized geo-routing algorithm. The traveller enters the intersection via the large blue arrow.

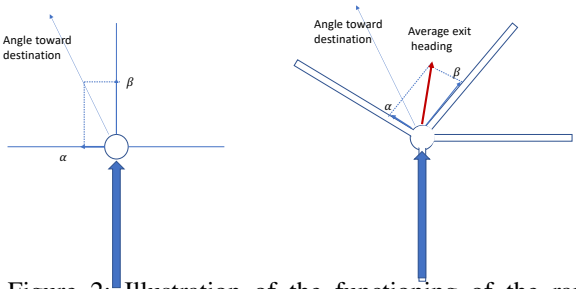

Figure 2: Illustration of the functioning of the randomized isotropic walk algorithm. The traveller enters the intersection via the large blue arrow.

III. Results of simulations of the algorithm

A. Simulation on a grid

We have chosen an abstraction of Paris map via a 100x100 map. Each grid point is an intersection in Paris. Each segment represents a street segment of one unit length. From North to South the streets represent 100 length units, and the total cumulative street length length is 20,000 length units. This compares well with the actual diameter of Paris (8km) and its cumulative street length of 1500 km (thus 2900 km of street sides).

The mayor of Paris has decided to create bike lanes parallel to the existing 14 subway lines, in order to foster the use of bikes instead of public transportation after city lock-down. To simulate those bike lanes we have highlighted 14 paths (seven paths West-East, seven paths North-South) on the grid map.

We have chosen $N_0 = 8,000$ source-destination pairs to run the two algorithms: the first one is the randomized algorithm, the second algorithm is the preferential path algorithm. In order to simulate the attraction effect of the highlighted path, we have artificially distorted the graph by decreasing the segment path weight by 20% and run a shortest path Dijkstra algorithm. This is a moderate ratio but it is sufficient to create a significant attraction effect, otherwise the difference would be not visible due to the large diversity of shortest paths in a grid structure. We simulate the isotropic randomized algorithm with $\epsilon = 0$.

Figures 3 and 4 show the histogram of the segment density. For each segment, we collect the traffic for both ways, and for all figures we add two traffic per segment to display the histogram. We notice that the preferential path algorithm shows that the preferential streets are drastically more busy than the other streets. The randomized algorithm shows a more balanced distribution of the traffic. The uniform isotropic condition fails on the borders of the map and holds in the large central part of the map.

Figure 5 shows the map of the traffic of street segments with the preferential path algorithm. In green we show the streets with load larger than 1 but smaller than 15, in blue - traffic smaller than 25, in red, smaller than 50, in black, smaller than 200. The preferential paths are clearly visible and marked as

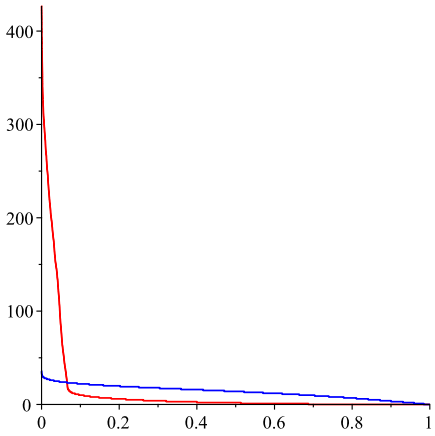

Figure 3: Histogram of segment traffic load in Grid Paris, in red the preferential path algorithm, in blue the randomized algorithm.

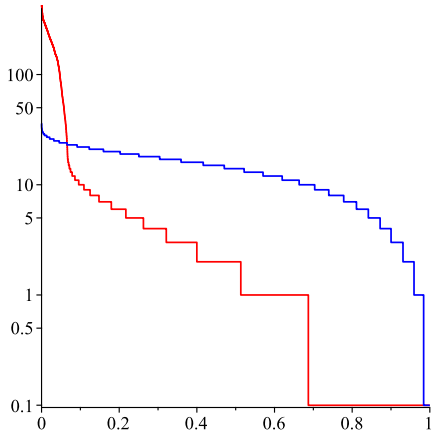

Figure 4: Histogram of segment densities in Grid Paris, in the logarithmic scale.

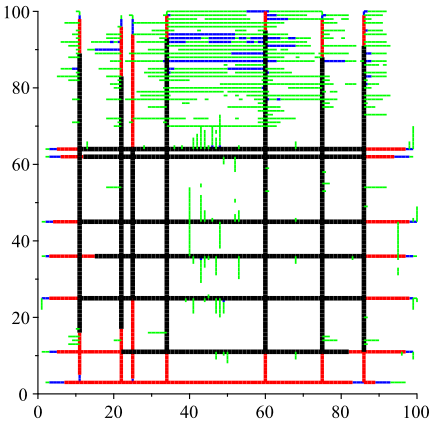

Figure 5: Map of segment densities in Grid Paris for the preferential path algorithm.

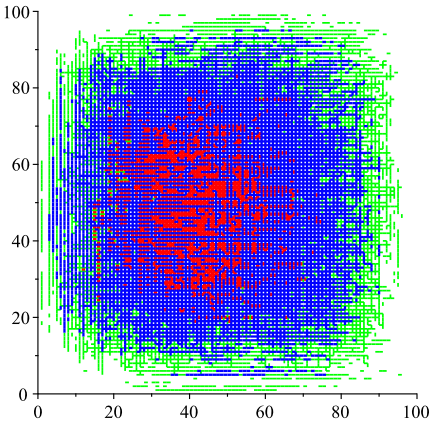

Figure 6: Map of segment densities in Grid Paris for the randomized algorithm.

very busy (black and red) as expected. Figure 6 shows the map of empty streets for the randomized algorithm, but the picture is now different, since the segment load is very balanced.

Now our aim is to derive the average exposure time of the travellers in Königsberg We took it to another scale and ran the algorithm on the whole network of recent city of Königsberg.

As in [1] we denote by $\mathcal{S}$ the set of street segments, with the difference, that the segments are directed and we differentiate a segment with its reverse segment. We denote $\ell(s)$ the segment length, thus the cumulative length is $L = \sum_{s\in\mathcal{S}} \ell(s)$. We denote $\lambda(s)$ the traffic load after N_0 initial-destination random pairs. The average path length is $L_G = \frac{1}{N_0}\sum_{s\in\mathcal{S}} \lambda(s)\ell(s)$. The average travel time is L_G/v.

If simulated time is T, on a segment s an entrance frequency rate is $\frac{\lambda(s)}{T}$ and the density on a segment is $\frac{\lambda(s)}{vT}$, assumed to be Poisson. If a number of travellers is N, then the density on a segment is $\frac{\lambda(s)}{vT}\frac{N}{N_0}$ where N_0 is a number of travellers needed to simulate an estimate $\lambda(s)$. Given the Poisson density, a probability that a random point at a random time on the segment does not have a traveller at distance within R_0 is $\exp\left(-\lambda(s)\frac{2R_0}{vT}\frac{N}{N_0}\right)$. We denoted R_0 as a safe distance against the virus, for biking $R_0 = 10m$. We write the expression of an average cumulative exposure time $E(N)$ for a random walk (traveller):

$$E(N) = \sum_{s\in\mathcal{S}} \frac{\lambda(s)\ell(s)}{vN_0}\left(1 - \exp\left(-\lambda(s)\frac{2R_0}{vT}\frac{N}{N_0}\right)\right). \quad (4)$$

If we want to know the exposure time $E_{12}(N)$ for a single traveller of the randomized algorithm, when all other travellers use the preferential paths, we get the expression

$$E_{12}(N) = \sum_{s\in\mathcal{S}} \frac{\lambda_2(s)\ell(s)}{vN_0}\left(1 - \exp\left(-\lambda_1(s)\frac{2R_0}{vT}\frac{N}{N_0}\right)\right) \quad (5)$$

where $\lambda_1(s)$ is the density of segment s for the preferential algorithm, and $\lambda_2(s)$ is the density for the randomized algorithm.

The Figures 7 and 8 show the average cumulative exposure time versus the travelling population in different situations: when all travellers are on preferential path, when all travellers are on the isotropic randomized algorithm, and the average cumulative exposure time when a single traveller is on the randomized algorithm, and the other are on preferential paths.

We display for a peak traffic period duration of 2 hours, and for a peak duration of 4 hours. For one day these quantities should be multiplied by two, since the travellers commute twice a day. For the traveller speed we have opted for $v = 12$km per hour, which is the average speed of bike commuting in Paris.

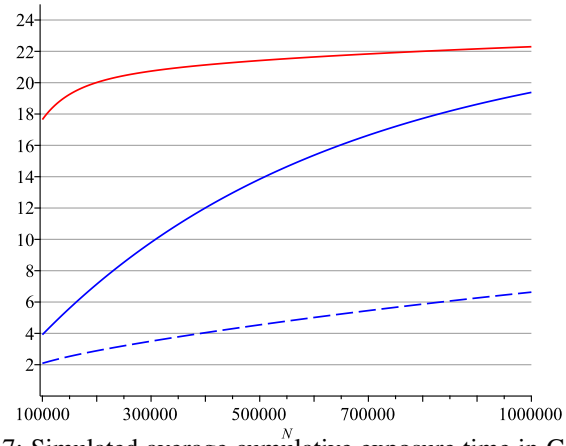

Figure 7: Simulated average cumulative exposure time in Grid Paris during a peak time of 2 hours versus total number of travellers: red: all traveller on preferential paths, blue: all travellers on randomized algorithm, dashed: one traveller on randomized algorithm. On x axis we show the number of commuters.

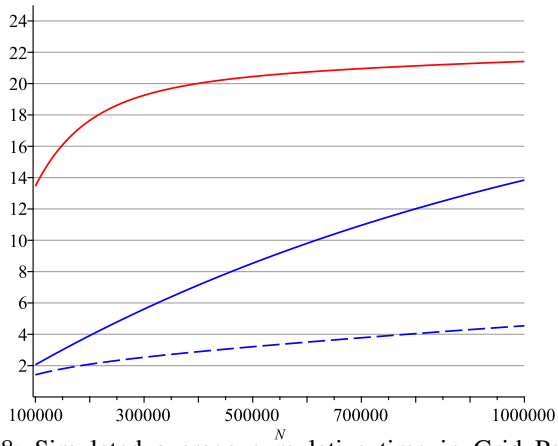

Figure 8: Simulated average cumulative time in Grid Paris during a peak time of 4 hours versus total number of travellers: red: all traveller on preferential path, blue: all travellers on randomized algorithm, dashed: one traveller on randomized algorithm. On x axis we show the number of commuters.

B. Simulation on a Voronoi triangulation

We have created a map of 4,000 random points on a 10kmx10km square and connected them through the Delaunay triangulation. In this simulation the streets are edges of a triangulation and nodes are intersections. With $4,000$ nodes we get a total street length of the order of 200 times the diameter of the map, thus showing a similar street density as Paris street map. As in the previous simulation we have connected 14 pairs of points, 7 North-South, seven West-East to simulate the subway network. The bike lane, which was supposed to follow the subway lines, is made of the path obtained via shortest path algorithm between the points of each pair.

In the preferential path algorithm we still use the shortest path algorithm but now we reduce the weight of the edges in the preferred path by 90%. This is kind of a drastic reduction but it simulates well the usage of subway where the commuter takes the path to the closest subway station and then changes to the subway station closest to his/her destination location.

We have run $N_0 = 5,000$ pairs of initial point and destination. Figures 9 and 10 show the histogram of the segment density loads consequence of the application of the algorithms. In red the preferential algorithm, in brown the isotropic walk algorithm, in blue the geo-routing algorithm, in green the shortest path algorithm. As expected the isotropic walk algorithm shows the best balance (better look at the logarithmic scale).

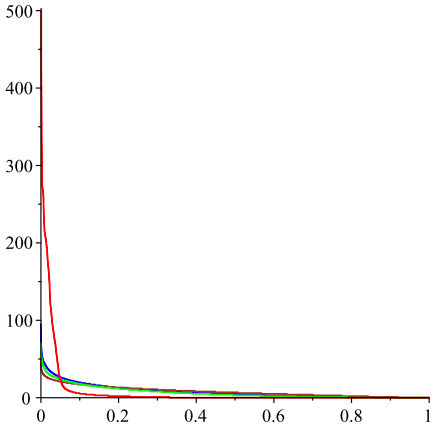

Figure 9: Histogram of segment traffic load in Delaunay Paris. In red we show the preferential path algorithm, in blue the randomized algorithm, in gree we show the streets with load.

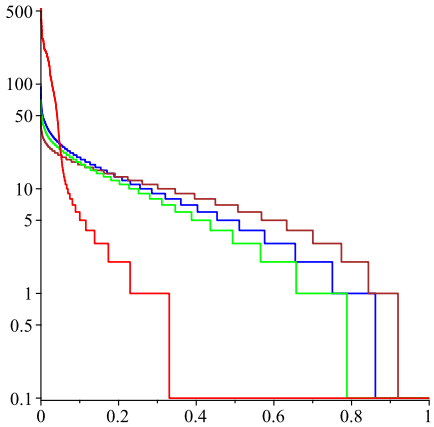

Figure 10: Histogram of segment densities in Delaunay Paris, in logarithmic scale. In red we show the preferential path algorithm, in blue the randomized algorithm.

Figure 11 shows the map of the traffic of street segments with the preferential path algorithm. In green we show the

streets with load larger than 1 but smaller than 15, in blue, traffic smaller than 25, in red, smaller than 50, in black, smaller than 200. The preferential paths are clearly visible and marked as very busy (black and red) as expected. Figure 12 shows the map of empty streets (lined in grey) forming a dense network for the shortest path algorithm, but the picture are now different, since the segment load are better balanced and the empty street much less dense. Figure 13 shows the same data for the isotropic walk path algorithm, since the segment load are very well balanced and the empty street very seldom. As expected the empty streets are on the border and the density balance more in the central part. Figure 14 shows the same data for the geo-routing path algorithm, since the segment load are a bit less balanced.

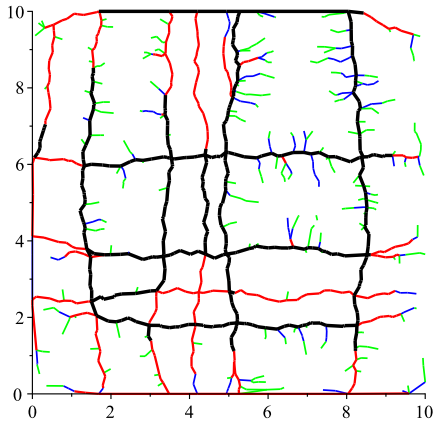

Figure 11: Map of segment densities in Delaunay triangulation of Paris for the preferential path algorithm.

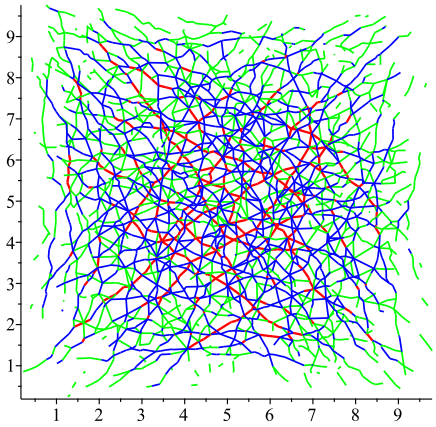

Figure 12: Map of segment densities in Delaunay Paris for the shortest path algorithm.

Figure 20 shows the cumulative exposure time experienced by the travellers in the different routing algorithms during a peak of 2 hours versus the commuting population size. The shortest path algorithm provides the smallest exposure because the average path length is shorter despite the path density is larger. Indeed the exposure rate discrepancy diminishes when the commuting population diminishes. But the shortest path

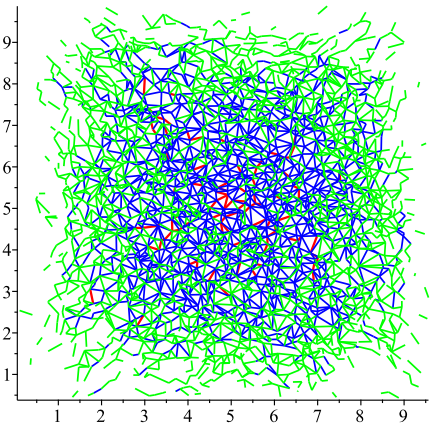

Figure 13: Map of segment densities in Delaunay Paris for the isotropic walk path algorithm.

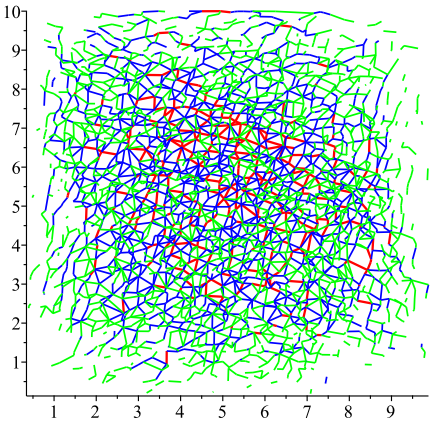

Figure 14: Map of segment densities in Delaunay Paris for the geo-routing path algorithm.

algorithm is too expensive in terms of complexity to answer to millions of request. The geo-routing and isotropic walk algorithms show similar performance. Although the isotropic algorithm benefits from its more balanced street loads at lower traffic. Figure 15 shows the respective average path length for each of the algorithms.

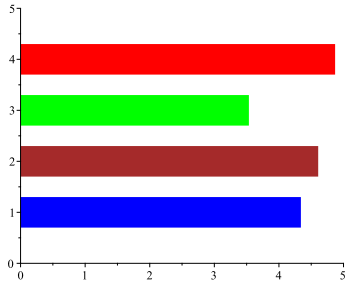

Figure 15: Average path length for four algorithms, in km, from top to bottom: preferential algorithm, shortest path algorithm, isotropic walk algorithm, geo-routing algorithm.

Since the preferential algorithm shows quite low performance, we investigate the possibility to multiply the parallel lanes to the preferential path in order to reduce the exposure

time via a reduced density. The Figure 17 shows the average exposure time versus the multiplicative factor of the parallel lanes. It turns out that the load on preferential lanes must be multiplied by at least factor of ten in order to get closer to the performance of the isotropic algorithm.

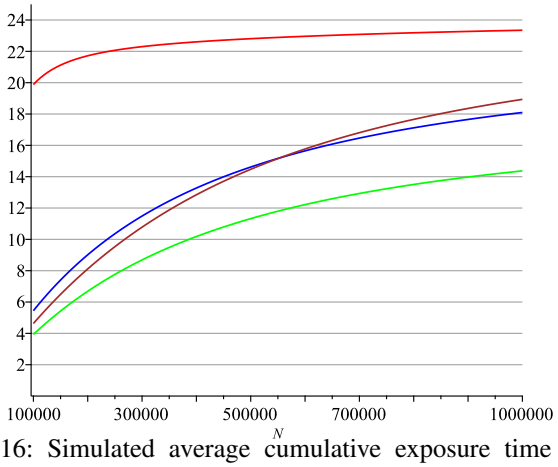

Figure 16: Simulated average cumulative exposure time in Delaunay Paris during a peak time of 2 hours versus total number of travellers: red: all traveller on preferential paths, green: all travellers on shortest path, brown the isotropic walk algorithm, blue the geo-routing algorithm.

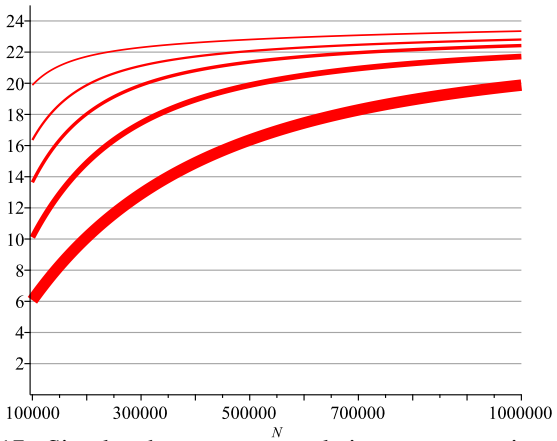

Figure 17: Simulated average cumulative exposure time in preferential path algorithm. From top to bottom: one lane, two parallel lanes, three parallel lanes, five parallel lanes, 10 parallel lanes.

C. Single pair simulation

In this section we evaluate the performance of the algorithms when we consider a single pair of an initial point and a destination point. We have simulated 1,000 travellers commuting between the two fixed points. The preferential path and shortest path algorithm are blocked on a single path because there is a unique solution to the Dijkstra algorithm. But for the geo-routing algorithm and the isotropic walk algorithms a stochastic component provides a diversity in the path choice. This way the two later algorithms offer a lower exposure thanks to the path diversity. Figures 18, 19 demonstrate the traffic of the segment for the four algorithms: yellow is for a traffic between 1 and 15 passage, green between 15 and 25, blue between 25 and 50, red between 50 and 200, black above 200. We notice that the path network of the geo-routing algorithm is more vascular and therefore uses less segments. Figure 20 shows the cumulative exposure time of the four algorithms in this situation.

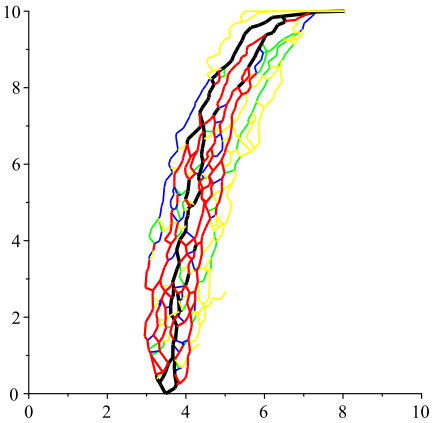

Figure 18: Map of segment densities in Delaunay Paris for the geo-routing path algorithm on a single pair.

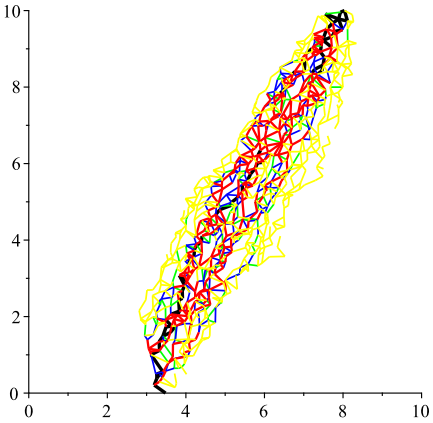

Figure 19: Map of segment densities in Delaunay Paris for the isotropic walk path algorithm on a single pair.

D. Single destination point or single initial point simulation

In this section we evaluate the performance of the algorithms when we consider a single destination or a single initial point. We have simulated 4,000 travellers commuting from or towards this point. The preferential path and shortest path algorithms indicate the same paths because there is a unique solution to the Dijkstra algorithm and the solution is symmetric in both ways. But for the geo-routing algorithm and the isotropic walk algorithms have a random component and are asymmetric with respect to the order in the pair made by the initial point and the destination point. Figures 23, 24 show the traffic on street segments for the two last algorithms when

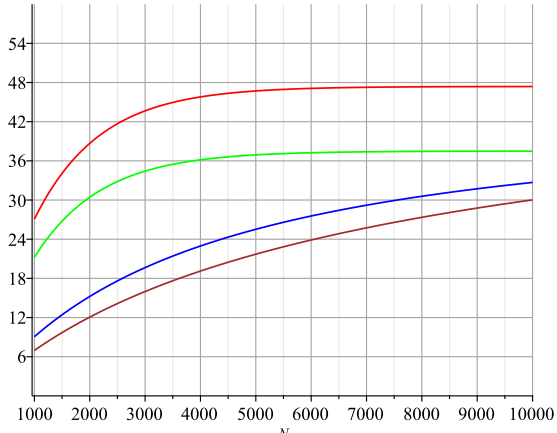

Figure 20: Simulated average cumulative exposure time in Delaunay Paris during a peak time of 2 hours versus total number of travellers on a single pair initial point and destination: red: all traveller on preferential paths, green: all travellers on shortest path, brown the isotropic walk algorithm, blue the geo-routing algorithm.

converging on the same destination point. Figure 21 shows the cumulative exposure time of four algorithms in this particular situation.

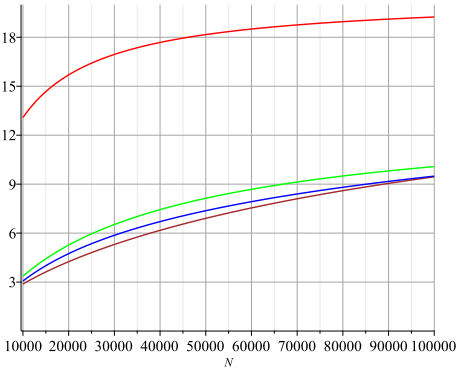

Figure 21: Simulated average cumulative exposure time in Delaunay Paris during a peak time of 2 hours versus total number of travellers on a single initial point. Red: all traveller on preferential paths, green: all travellers on shortest path, brown the isotropic walk algorithm, blue the geo-routing algorithm.

E. The advanced algorithm over a real network of a city

We have made simulations on real city maps. The interesting part is when the road maps shows local complication, such as obstacles, river, lakes, etc. In this case the routing algorithm may loop for ever in some areas. We have chosen the city of Königsberg.

In order to cope with looping problems of the algorithm we used the concept of the so-called accessibility graph, also used in [6]. Let $G = G(V, E)$ be a graph with vertex set V and edge set E. The second degree accessibility graph of the

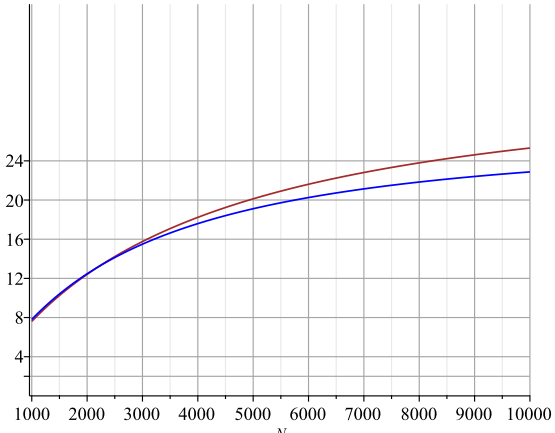

Figure 22: Simulated average cumulative exposure time in Delaunay Paris during a peak time of 2 hours versus total number of travellers on a fixed pair of initial point and destination: brown the isotropic walk algorithm, blue the geo-routing algorithm.

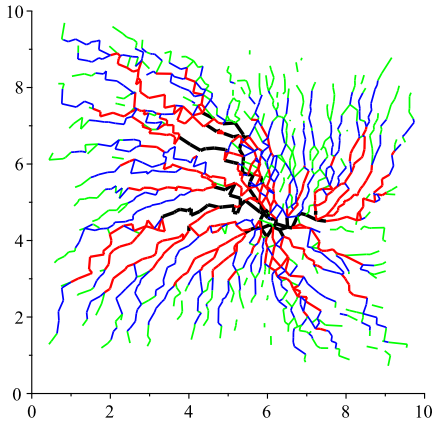

Figure 23: Map of segment densities in Delaunay Paris for the geo-routing path algorithm converging on a single destination point.

graph $G * G$ or G^{*2} is the graph whose vertex set is V and the edge set is E_2 such that for $(x, y) \in V^2$, $(xy) \in E_2$ if there exists $z \in V$ such that (xz) and (zy) belong to E. In other words G^{*2} is the "two hop" graph of G.

We have computed G^{*2}, G^{*4} and G^{*8}. If the algorithm on a path determination loops and eventually fails on G, we run the algorithm on G^{*2}. Then, in general, if it fails on G^{*2^k} we run the algorithm on the accessibility graph of the next degree $G^{*2^{k+1}}$ until the accessibility graph becomes fully connected graph. When the path is determined one unfold the path until we get a path on an original graph G.

Figure 26 shows the street density with the geo-routing algorithm using the accessibility graph option. Figure 27 demonstrates the street density when the source is fixed in the center of the city. Figure 28, when the source and destination are both fixed. The performance of the algorithms is estimated by comparing with the original graph G, where in 95% cases the algorithm fail due to the complexification of the map, but

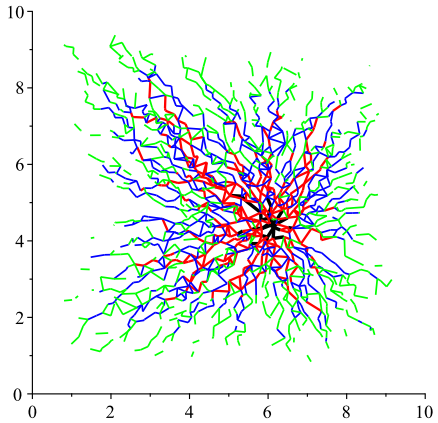

Figure 24: Map of segment densities in Delaunay Paris for the isotropic walk path algorithm converging on a single destination point.

Figure 25: Map of Königsberg (now Kaliningrad) city, made using openstreetmap.

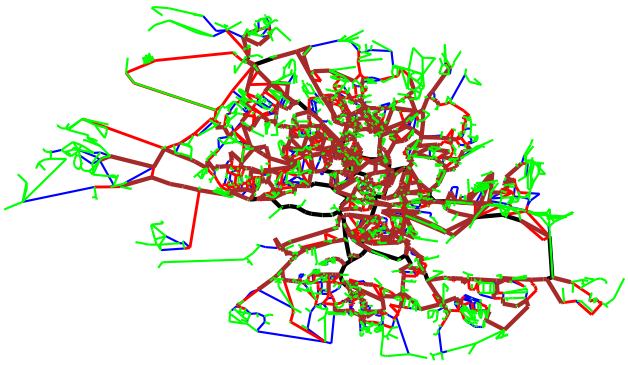

Figure 26: Map of Königsberg street density for the geo-routing algorithm when sources and destinations are chosen at random.

on the accessibility graph G^{*2} 39% only of cases fail, on G^{*4}, 6%, and on G^{*8} no fail.

IV. Conclusion

The epidemic outbreak of COVID-19 has led to exceptional lock-down measures in most countries. Critical situation happened after lock-downs and travel restrictions [8] in cities of many countries around the globe. In many cases the exposure to virus occurs during outdoor excursions in the city when commuting to work or shopping. A special care is given when the excursions are made by bike in order to avoid both, road congestion and public transportation virus exposure. In some cities such as Paris special preferential biking lanes has been opened to follow more or less the trajectories of public transportation. Unfortunately the algorithms of transport regularisation based on preferential path idea may lead to increase of the virus exposure. In this work we presented path selection algorithms generalised from the initial idea presented in [1] to reduce virus exposure. As the result of numerical simulations of the algorithms we see that the path reduce exposure time to the virus by a factor ranging from 3 to 9. The algorithms are fully distributed and are simple to run. In the manuscript we performed the algorithm on two different types of models: grid-like models of cities (Paris grid) and non-grid networks of real cities like Königsberg.

We believe that mathematical applications of algorithms, such as Ariadne may help to improve surveillance strategies for preventing epidemics on a local scale of a city. Furthermore when specifically adapting such algorithms to the epidemiological context of models and data [4], [7], [11] can lead to development of algorithms with real world applications. As an outlook of this work we aim to also implement the algorithm also using the open data [10] of real number of cases in the city and include this information to the network of the city.

References

[1] P. Jacquet, "The Ariadne String against Covid-19 pandemic propagation during lock-downs". https://hal.archives-ouvertes.fr/hal-02546347/ (2020)

[2] P. Jacquet, B. Mans, P. Muhlethaler, & G. Rodolakis, "Opportunistic routing in wireless ad hoc networks: Upper bounds for the packet propagation speed". IEEE Journal on Selected Areas in Communications, 27(7), 1192-1202 (2009)

[3] P. Jacquet, S. Malik, B. Mans, & A. Silva, "On the throughput-delay trade-off in georouting networks" IEEE Transactions on Information Theory, 62(6), 3230-3242 (2016)

[4] E. Colman, P. Holme, H. Sayama and C. Gershenson, "Efficient sentinel surveillance strategies for preventing epidemics on networks", PLOS Comp. Biol. 15, e1007517 (2019)

[5] N. Oliver, B. Lepri, H. Sterly, R. Lambiotte, S. Deletaille, M.de Nadai, E. Letouze, A. A. Salah, R. Benjamins, C. Cattuto, V. Colizza, N. de Cordes, S.P. Fraiberger, T. Koebe, S.Lehmann, J. Murillo, A. Pentland, P. N Pham, F. Pivetta, J. Saramäki, S. V. Scarpino, M. Tizzoni, S. Verhulst and P. Vinck, "Mobile phone data for informing public health actions across the COVID-19 pandemic life cycle", Vol 6, No. 23, Sci. Adv. (2020)

[6] H.H.K. Lentz, T. Selhorst, I.M. Sokolov "Unfolding accessibility

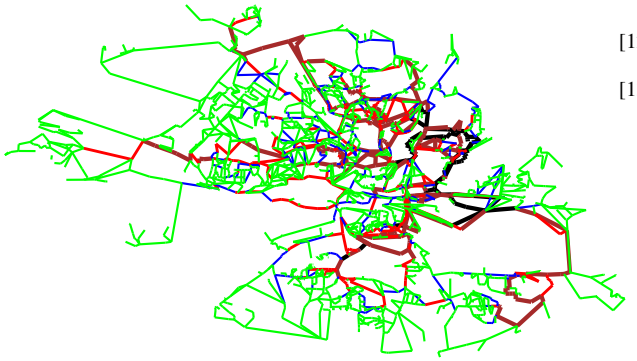

Figure 27: Map of Königsberg street density for the **geo-routing** algorithm when source is fixed and destination is chosen **at random**.

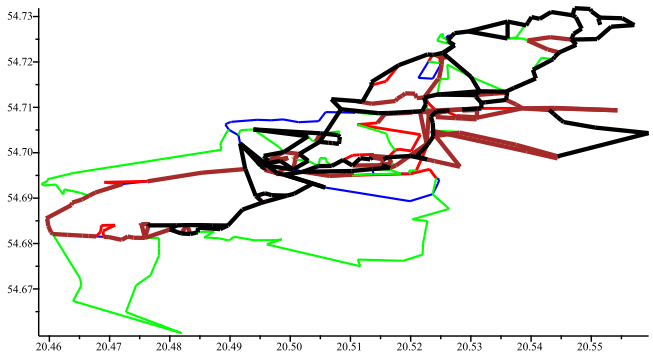

Figure 28: Map of Königsberg street density for the **geo-routing** algorithm when source and destination **are fixed**.

provides a macroscopic approach to temporal networks" Physical review letters 110 (11), 118701 (2013)

[7] G. Pullano, F. Pinotti, E. Valdano, P.-Y. Boëlle, C. Poletto, V. Colizza "Novel coronavirus (2019-nCoV) early-stage importation risk to Europe, January 2020", 25, 4, Eurosurveillance (2020)

[8] B. Klein, T. LaRock, S. McCabe, L. Torres, L. Friedland, F. Privitera, B. Lake, M. U. G. Kraemer, J. S. Brownstein, D. Lazer, T. Eliassi-Rad, S. V. Scarpino, A. Vespignani, and M. Chinazzi "Reshaping a nation : Mobility, commuting, and contact patterns during the COVID-19 outbreak", report https://www.mobs-lab.org/uploads/6/7/8/7/6787877/covid19mobility_report2.pdf (2020)

[9] M. Chinazzi, J. T. Davis, M. Ajelli, C. Gioannini, M. Litvinova, S. Merler, A. Pastore y Piontti, K. Mu, L. Rossi, K. Sun, C. Viboud, X. Xiong, H. Yu, M. Elizabeth Halloran, I. M. Longini Jr., A. Vespignani, "The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak", Sci. Adv.2020;6: eabc076 (2020)

[10] COVID database report https://covid19mm.github.io/in-progress/2020/06/19/sixth-report.html

[11] F. Schlosser, B. F. Maier, D. Hinrichs, A. Zachariae, D. Brockmann, "COVID-19 lockdown induces structural changes in mobility networks – Implication for mitigating disease dynamics", https://arxiv.org/abs/2007.01583

[12] L. Dumaz, "How to flee along a straight line: Tracking Self-Repelling Random Walks", Prisme N31 (2015)

[13] Openstreetmap data available at openstreetmap.org